REMARKS/ARGUMENTS

Applicant respectfully requests reconsideration and allowance of the subject application.

Claims 1-38 were originally submitted.

Claims 1, 9, 15, and 21 were previously amended.

Claims 1-38 remain in this application.

35 U.S.C. §102

Claims 1, 3, 4, 6-13, 15-18, 20-23, 25, 26, 28, 29, and 31-37 are rejected under 35 U.S.C. 102(b) as being anticipated by US Patent No. 6,430,576 to Gates et al (Gates). Applicant respectfully traverses the rejection.

Gates discloses that in the conventional systems, to control and execute certain applications, information etc. an object is used, where the object refers to an entity that contains data and the associated functions that can use or affect that data. The object is implemented through an object oriented programming. A request is sent to the object to execute the functions.

Gates describes a method and an apparatus to synchronize the object that is copied from a remote location to a local space at a client to match the running applications. The required synchronization can be with respect to time, type of request to be executed or the number of requests to be processed. Gates further describes a policy that specifies the conditions for synchronizing the object. The policy can be a default policy or based upon the user action. The rule set of the policy can also be determined by analyzing the requests exchanged between the server and the client. The synchronization of the copy of the object located at the client side with the copy of the object present at the remote location is carried out,

where the synchronization is performed by transmitting minimal information about any change in the local copy of the object from the client side to a remote system such as server. The function of transmitting the minimal information is managed by the editing context. Consequently, the updates in the copies of the object present at other remote locations are made by the server.

The object can also have a private data with access limited to a particular client. Therefore, only the non-confidential information is passed while creating the object for the client having limited or no access to the private data. The object can also be retrieved from the server for creating the required copy of the object at the client side by using certain segments of the object that have not been previously used. This method of object creation is called as faulting. The segment can contain the reference to other objects at some remote location from which the data is needed to be retrieved. For retrieval of objects, the method uses an object graph editing context. The object graph provides a list of objects that accessible by a particular client. The editing context helps in getting only the required data and not the whole of information contained in the referenced object. The editing context at the client is called as object store. The required data included in the referenced object is acquired only when it is absent from the local space at the client. The editing context is also used to manage the local object graph. To access the remote or referenced objects, a object persistent mechanism is used. In this mechanism, an object store sends a request to the server editing context. The server editing context provides for the connection to the external facility such as a relational database or a file system where the object is actually persistent or present.

Independent claim 1, for example, recites "[a] method comprising:

> generating a policy digest for a cached policy that applies to a client, the policy digest identifying at least one assertion the client is complying with; and
>
> including the policy digest in a request by the client to access a resource.

Gates does not disclose the element "generating a policy digest for a cached policy that applies to a client" as recited by claim 1. In contrast, Gates discloses distribution and synchronization of copies of objects locally. The policy responsible for synchronization of objects is based upon specific conditions such as time, processing of the type of messages, processing of the number of messages, etc. The conditions or rules for the policy are determined by evaluating the messages exchanged between the local object and the remote object. See Gates, Abstract; col. 1, lines 21-25 and 62-66; col. 3, lines 56-66; and col. 6, lines 36-44 and 61-67.

For example, the Application describes generating a policy digest for a cached policy that applies to a client. A client messaging module is configured to generate the policy digest. The policy digest includes the assertions (e.g., rules and declarations) which govern access to the resources and policy identity which identifies the cached policy used to communicate with the host (i.e., server). The cached policy is applied through the policy digest with the help of associated assertions and the policy identity. The policy digest checks for the validity of the policy when the policy digest in a request message is transmitted from the client to the server to access a resource at the server.

Accordingly, Gates fails to disclose or show each recited element of claim 1, and the rejection is improper. Applicant respectfully requests that the §102 rejection of claim 1 be withdrawn.

Dependent claims 3, 4, 6-8 depend on claim 1, and are allowable at the least by virtue of their dependency on base claim 1. Accordingly, Applicant respectfully request that the §102 rejection of claims 3, 4, 6-8 be withdrawn based on the reasons provided in support of claim 1. Furthermore, particular dependent claims are allowable based on additional reasons provided below.

Claim 3 further recites "[t]he method of claim 1, wherein generating the policy digest includes encoding a bit vector identifying selected assertions from the cached policy." Gates does not disclose or show the element of :generating a policy digest includes encoding a bit vector identifying selected assertions from the cached policy" as recited by claim 3. Dates discloses that the synchronization of the object that is copied from a remote location to a local space at a client to match the running applications. The required synchronization can be with respect to time, type of request to be executed or the number of requests to be processed. Gates describes a policy that specifies the conditions for synchronizing the object. The policy can be a default policy or based upon the user action. The rule set of the policy can also be determined by analyzing the requests exchanged between the server and the client. The synchronization of the copy of the object located at the client side with the copy of the object present at the remote location is carried out. This synchronization is performed by transmitting minimal information about any change in the local copy of the object from the client side to a remote system such as server. The function of transmitting the minimal information is managed by the editing context. Consequently, the updates in the copies of the object present at

other remote locations are made by the server. The object can also have a private data with access limited to a particular client. Hence, only the non-confidential information is passed while creating the object for the client having limited or no access to the private data. See Gates, col. 6, lines 36-60, col. 7, lines 50-61.

For example, the Application discloses generating a policy digest includes encoding a bit vector identifying selected assertions from the cached policy. The bit vector written in XML in the cached policy is encoded with the policy digest. The binary value of the bit vector is encoded as a text value indicating the assertion selected in the policy digest. The bit value 1 indicates selection of a particular assertion and the bit value 0 refers to the non-selection of the particular assertion. In contrast, Gates discloses that the object is mutated or transformed into a different kind of object containing only some specific information which is allowed to be accessible by the client. See Gates, col. 7, lines 50-57.

Claim 6 further recites "[t]he method of claim 1, further comprising: incrementing a counter each time the cached policy is used; and removing the cached policy from a cache at the client when the counter exceeds a limit value."

Gates does not disclose or show the element "incrementing a counter each time the cached policy is used; and removing the cached policy from a cache at the client when the counter exceeds a limit value" as recited by claim 6. Gates discloses the synchronization of the object that is copied from a remote location to a local space at a client to match the running applications. The required synchronization can be with respect to time, type of request to be executed or the number of requests to be processed. Gates describes a policy that specifies the conditions for synchronizing the object. The policy can be a default policy or based upon the user action. The rule set of the policy can also be determined by

analyzing the requests exchanged between the server and the client. The synchronization of the copy of the object located at the client side with the copy of the object present at the remote location is carried out. This synchronization is performed by transmitting minimal information about any change in the local copy of the object from the client side to a remote system such as server. The function of transmitting the minimal information is managed by the editing context. Consequently, the updates in the copies of the object present at other remote locations are made by the server. See Gates, col. 6, lines 36-60.

For example, the Application discloses a counter which is associated with the cached policy. The value of the counter is updated when the cached policy located at the client side is used for comparison with the policy located at the server. When the policy at the client side does not match with the policy that is located at the server, then the value of the counter is incremented. The policy at the client side is also removed, when the value of the counter exceeds a limiting value.

Claim 7 further recites "[t]he method of claim 1, further comprising: incrementing a counter for the cached policy when a fault is received at the client in response to using the cached policy; and removing the cached policy from a cache at the client when the counter exceeds a limit value."

Gates does not disclose or show the element "incrementing a counter for the cached policy when a fault is received at the client in response to using the cached policy; and removing the cached policy from a cache at the client when the counter exceeds a limit value" as recited by claim 7.

Gates discloses the synchronization of the object that is copied from a remote location to a local space at a client to match the running applications. The

required synchronization can be with respect to time, type of request to be executed or the number of requests to be processed. Gates describes a policy that specifies the conditions for synchronizing the object. The rule set of the policy can also be determined by analyzing the requests exchanged between the server and the client. The object can also be retrieved from the server for creating the required copy of the object at the client side by using certain segments of the object that have not been previously used. The method of object creation is known as faulting. The segment can contain the reference to other objects at some remote location from which the data is needed to be retrieved. For retrieval of objects, the object graph editing context helps in getting only the required data and not the whole of information contained in the referenced object. The required data included in the referenced object is acquired only when it is absent from the local space at the client. See Gates, col. 6, lines 36-60 and col. 9, lines 32-48.

For example, the Application discloses incrementing a counter for the cached policy when a fault is received at the client in response to using the cached policy; and removing the cached policy from a cache at the client when the counter exceeds a limit value. The value of the counter is incremented when a fault is received from the server. The fault is received when the policy located at the client side does not match with the policy that is located at the server. The policy at the client side is removed when the value of the counter due to faults exceeds a limiting value.

Claim 8 further recites "[t]he method of claim 1, further comprising logging a diagnostic event when a fault is received at the client to identify a system problem."

Gates does not disclose or show the element "logging a diagnostic event when a fault is received at the client to identify a system problem" as recited by claim 8. Gates discloses that the object can be retrieved from the server for creating the required copy of the object at the client side by using certain segments of the object that have not been previously used. The method of object creation is known as faulting. The segments contain reference to other objects at some remote location from which the data is required to be retrieved. For retrieval of objects, the object graph editing context is used that helps in getting only the required data and not the whole of information contained in the referenced object. The editing context at the client is called as object store. The required data included in the referenced object is acquired only when it is absent from the local space at the client. The editing context is also used to manage the local object graph. To access the remote or referenced objects, a object persistent mechanism is used. In this mechanism, an object store sends a request to the server editing context. The server editing context provides for the connection to the external facility such as a relational database or a file system where the object is actually persistent or present. See Gates, col. 9, lines 49-67.

For example, the Application discloses logging of a diagnostic event when a fault is received at the client to identify a system problem. The fault triggers a diagnostic event at the client which identifies the problem with the system and maintains a log for administrator's review.

Independent claim 9 recites the element "extracting at a host a policy digest identifying a cached policy that applies to a client, the policy digest included in a request to access a resource".

As discussed above in support of claim 1, Gates discloses distribution and synchronization of copies of objects locally. The policy responsible for synchronization of objects is based upon specific conditions such as time, processing of the type of messages, processing of the number of messages, etc. The conditions or rules for the policy are determined by evaluating the messages exchanged between the local object and the remote object. See Gates, Abstract; col. 1, lines 21-25 and 62-66; col. 3, lines 56-66; and col. 6, lines 36-44 and 61-67.

Accordingly, Gates fails to disclose or show each recited element of claim 9, and the rejection is improper. Applicant respectfully requests that the §102 rejection of claim 9 be withdrawn.

Dependent claims 10-13 depend on claim 9, and are allowable at the least by virtue of their dependency on base claim 9. Accordingly, Applicant respectfully request that the §102 rejection of claims 10-13 be withdrawn based on the reasons provided in support of claim 9. Furthermore, particular dependent claims are allowable based on additional reasons provided below.

Claim 12 further recites "[t]he method of claim 9, further comprising decoding a bit vector of the cached policy."

Gates does not disclose or show the element "decoding a bit vector of the cached policy" as recited by claim 12. Gates discloses the synchronization of the copy of the object located at the client side with the copy of the object present at the remote location. This is performed by transmitting minimal information about any change in the local copy of the object from the client side to a remote system such as server. Therefore, the updates in the copies of the object present at other remote locations are made by the server. See Gates, col. 6, lines 61-67, col. 7, lines 15-24.

For example, the Application discloses decoding of a bit vector that corresponds to a particular assertion mentioned in the policy; however Gates discloses that the object is mutated or transformed into a different kind of object containing only some specific information which is allowed to be accessible by the client. See Gates, col. 7, lines 50-57.

Independent claim 15 recites the element "a policy digest identifying at least one cached policy that applies to a client". Applicant presents arguments presented in support of claim 1, in support of claim 15.

Accordingly, Gates fails to disclose or show each recited element of claim 15, and the rejection is improper. Applicant respectfully requests that the §102 rejection of claim 15 be withdrawn.

Dependent claims 16-18 and 20 depend on claim 15, and are allowable at the least by virtue of their dependency on base claim 15. Accordingly, Applicant respectfully request that the §102 rejection of claims 16-18 and 20 be withdrawn based on the reasons provided in support of claim 15. Furthermore, particular dependent claims are allowable based on additional reasons provided below.

Claim 18 further recites "[t]he system of claim 15, wherein the policy digest is a bit vector of a cached policy."

Gates does not disclose or show "the policy digest is a bit vector of a cached policy" as recited in claim 18. Gates discloses the synchronization of the object that is copied from a remote location to a local space at a client to match the running applications. The required synchronization can be with respect to time, type of request to be executed or the number of requests to be processed. Gates describes a policy that specifies the conditions for synchronizing the object. The policy can be a default policy or based upon the user action. The rule set of the

policy can also be determined by analyzing the requests exchanged between the server and the client. The synchronization of the local copy of the object (client side) with the copy present at the remote location (server side) is performed by transmitting minimal information about any change in the local copy of the object from the client side to a remote system such as server. The function of transmitting the minimal information is managed by the editing context. Further, the updates in the copies of the object present at other remote locations are performed by the server. The object can also have a private data with access limited to a particular client. Therefore, only the non-confidential information is passed while creating the object for the client having limited or no access to the private data. See Gates, col. 6, lines 36-60, col. 7, lines 50-61.

For example, the Application discloses that the policy digest is a bit vector for a cached policy. The bit vector can have a binary value, either 1 or 0. The bit vector is encoded as a text value indicating the assertion selected in the policy digest. The bit value 1 indicates the selection of a particular assertion and the bit value 0 refers to the non-selection of the particular assertion. However, Gates discloses that the object mutation or transformation into a different kind of object containing only some specific information which is allowed to be accessible by the client. The rule set of the policy can also be determined by analyzing the requests exchanged between the server and the client.

Independent claim 21, recites "[a] system comprising:

> a policy digest for a cached policy that applies to a client, the policy digest identifying at least one assertion the client is complying with; and
>
> a messaging module including the policy digest in a request by the client to access a resource.

Gates does not disclose or show the element "a policy digest for a cached policy that applies to a client, the policy digest identifying at least one assertion the client is complying with" as recited by claim 21. Gates discloses distribution and synchronization of copies of objects locally. The policy responsible for synchronization of objects is based upon specific conditions such as time, processing of the type of messages, processing of the number of messages, etc. The conditions or rules for the policy are determined by evaluating the messages exchanged between the local object and the remote object. See Gates, Abstract; col. 1, lines 21-25 and 62-66; col. 3, lines 56-66; col. 6, lines 36-44 and 61-67.

For example, the Application discloses a policy digest for a cached policy that applies to a client, the policy digest identifying at least one assertion the client is complying with. The client messaging module is configured to generate the policy digest. The policy digest includes the assertions (e.g., rules and declarations) which govern access to the resources and policy identity which identifies the cached policy used to communicate with the host (i.e., server). The cached policy is applied through the policy digest with the help of associated assertions and the policy identity. The basic purpose of the policy digest is to check for the validity of the policy, when the policy digest in a request message is transmitted from the client to the server to access a resource at the server.

Accordingly, Gates fails to disclose or show each recited element of claim 21, and the rejection is improper. Applicant respectfully requests that the §102 rejection of claim 21 be withdrawn.

Dependent claims 22, 23 and 25 depend on claim 21, and are allowable at the least by virtue of their dependency on base claim 21. Accordingly, Applicant respectfully request that the §102 rejection of claims 22, 23 and 25 be withdrawn

based on the reasons provided in support of claim 21. Furthermore, particular dependent claims are allowable based on additional reasons provided below.

Claim 23 further recites "[t]he system of claim 21, wherein the policy digest is a bit vector of a cached policy." Claim 23 benefits from arguments presented in support of claim 18.

Independent claim 26 recites the element "generating a policy digest for a cached policy that applies to a client, the policy digest identifying at least one assertion the client is complying with". Applicant presents arguments presented in support of claim 1, in support of claim 26.

Accordingly, Gates fails to disclose or show each recited element of claim 26, and the rejection is improper. Applicant respectfully requests that the §102 rejection of claim 26 be withdrawn.

Dependent claims 28, 29, and 31-33 depend on claim 26, and are allowable at the least by virtue of their dependency on base claim 26. Accordingly, Applicant respectfully request that the §102 rejection of claims 28, 29, and 31-33 be withdrawn based on the reasons provided in support of claim 26. Furthermore, particular dependent claims are allowable based on additional reasons provided below.

Claim 28 further recites "[t]he computer program product of claim 26 wherein the computer process further comprises encoding a bit vector of the cached policy." Claim 23 benefits from arguments presented in support of claim 12.

Claim 29 further recites "[t]he computer program product of claim 26 wherein the computer process further comprises reading an assertion from the

policy, assigning a bit value to the assertion, and writing the bit value to a bit vector."

Gates does not disclose or show "reading an assertion from the policy, assigning a bit value to the assertion, and writing the bit value to a bit vector" as recited in claim 29. Gates discloses the synchronization of the object that is copied from a remote location to a local space at a client to match the running applications. The required synchronization can be with respect to time, type of request to be executed or the number of requests to be processed. Gates further describes a policy that specifies the conditions for synchronizing the object. The policy can be a default policy or based upon the user action. The rule set of the policy can also be determined by analyzing the requests exchanged between the server and the client. The synchronization of the copy of the object located at the client side with the copy of the object present at the remote location is carried out. This synchronization is performed by transmitting minimal information about any change in the local copy of the object from the client side to a remote system such as server. The function of transmitting the minimal information is managed by the editing context. Consequently, the updates in the copies of the object present at other remote locations are made by the server. The object can also have a private data with access limited to a particular client. Therefore, only the non-confidential information is passed while creating the object for the client having limited or no access to the private data. See Gates, col. 6, lines 36-60, col. 7, lines 50-61.

For example, the Application discloses reading an assertion from the policy, assigning a bit value to the assertion, and writing the bit value to a bit vector The bit vector written in XML in the cached policy is encoded with the policy digest. The binary value of the bit vector is encoded as a text value indicating the

assertion selected in the policy digest. The bit value 1 indicates selection of a particular assertion and the bit value 0 refers to the non-selection of the particular assertion. However Gates discloses that the object is mutated or transformed into a different kind of object containing only some specific information which is allowed to be accessible by the client. See Gates, col. 7, lines 50-57.

Claim 31 further recites "[t]he computer program product of claim 26, wherein the computer process further comprises: incrementing a counter each time the cached policy is used; and removing the cached policy from a cache at the client when the counter exceeds a limit value."

Gates does not disclose or show "incrementing a counter each time the cached policy is used; and removing the cached policy from a cache at the client when the counter exceeds a limit value" as recited in claim 31. Gates discloses the synchronization of the object that is copied from a remote location to a local space at a client to match the running applications. The required synchronization can be with respect to time, type of request to be executed or the number of requests to be processed. Gates describes a policy that specifies the conditions for synchronizing the object. The policy can be a default policy or based upon the user action. The rule set of the policy can also be determined by analyzing the requests exchanged between the server and the client. The synchronization of the copy of the object located at the client side with the copy of the object present at the remote location is carried out. This synchronization is performed by transmitting minimal information about any change in the local copy of the object from the client side to a remote system such as server. The function of transmitting the minimal information is managed by the editing context. Consequently, the updates in the

copies of the object present at other remote locations are made by the server. See Gates, col. 6, lines 36-60.

For example, the Application discloses a counter which is associated with the cached policy. The value of the counter is updated when the cached policy located at the client side is used for comparison with the policy located at the server. When the policy at the client side does not match with the policy that is located at the server, then the value of the counter is incremented. The policy at the client side is also removed, when the value of the counter exceeds a limiting value.

Claim 33 further recites "[t]he computer program product of claim 26 wherein the computer process further comprises triggering a diagnostic event when a fault is received at the client."

Gates does not disclose or show "triggering a diagnostic event when a fault is received at the client" as recited by claim 33. Gates discloses that the object can be retrieved from the server for creating the required copy of the object at the client side by using certain segments of the object that have not been previously used. The method of object creation is known as faulting. The segments contain reference to other objects at some remote location from which the data is required to be retrieved. For retrieval of objects, the object graph editing context is used that helps in getting only the required data and not the whole of information contained in the referenced object. The editing context at the client is called as object store. The required data included in the referenced object is acquired only when it is absent from the local space at the client. The editing context is also used to manage the local object graph. To access the remote or referenced objects, a object persistent mechanism is used. In this mechanism, an object store sends a

request to the server editing context. The server editing context provides for the connection to the external facility such as a relational database or a file system where the object is actually persistent or present. See Gates, col. 9, lines 49-67.

For example, the Application discloses triggering a diagnostic event when a fault is received at the client. The fault triggers a diagnostic event at the client which identifies the problem with the system and maintains a log for administrator's review.

Independent claim 34 recites the element "extracting at a host a policy digest identifying a cached policy that applies to a client". Applicant presents arguments presented in support of claim 1, in support of claim 34.

Accordingly, Gates fails to disclose or show each recited element of claim 34, and the rejection is improper. Applicant respectfully requests that the §102 rejection of claim 34 be withdrawn.

Dependent claims 35-37 depend on claim 34, and are allowable at the least by virtue of their dependency on base claim 34. Accordingly, Applicant respectfully request that the §102 rejection of claims 35-37 be withdrawn based on the reasons provided in support of claim 34. Furthermore, particular dependent claims are allowable based on additional reasons provided below.

Claim 36 further recites "[t] computer program product of claim 34 wherein the computer process further comprises decoding a bit vector of the cached policy." Claim 36 benefits from arguments presented in support of claim 12.

35 U.S.C. §103

Claims 2, 5, 14, 19, 24, 27, 30, and 38 are rejected under 35 U.S.C 103(a) as being unpatentable over Gates, as applied to claim 1, 9, 15, 21, 26 and 34, in further view of U.S. Patent No. 6,519,764 issued to Atkinson et al (Atkinson).

Dependent claims 2 and 5 depend on claim 1, and include all the elements of claim 1. Gates, as discussed above in support of claim 1, fails to teach or suggest the element "generating a policy digest for a cached policy that applies to a client". Atkinson is cited by the Action as to generating or using a hash of the policy; however, Atkinson is of no assistance in light of the teachings of Gates.

Accordingly, Gates in view of Atkinson fails to teach or suggest each recited element of claims 2 and 5, and the rejection is improper. Applicant respectfully requests that the §103 rejection of claims 2 and 5 be withdrawn.

Dependent claim 14 depends on claim 9, and includes all the elements of claim 9. Gates, as discussed above in support of claim 9, fails to teach or suggest the element "extracting at a host a policy digest identifying a cached policy that applies to a client, the policy digest included in a request to access a resource"; however, Atkinson is of no assistance in light of the teachings of Gates.

Accordingly, Gates in view of Atkinson fails to teach or suggest each recited element of claim 14, and the rejection is improper. Applicant respectfully requests that the §103 rejection of claim 14 be withdrawn.

Dependent claim 19 depends on claim 15, and includes all the elements of claim 15. Gates, as discussed above in support of claim 15, fails to teach or suggest the element "a policy digest identifying at least one cached policy that applies to a client"; however, Atkinson is of no assistance in light of the teachings of Gates.

Accordingly, Gates in view of Atkinson fails to teach or suggest each recited element of claim 19, and the rejection is improper. Applicant respectfully requests that the §103 rejection of claim 19 be withdrawn.

Dependent claim 24 depends on claim 21, and includes all the elements of claim 21. Gates, as discussed above in support of claim 21, fails to teach or suggest the element "a policy digest for a cached policy that applies to a client, the policy digest identifying at least one assertion the client is complying with"; however, Atkinson is of no assistance in light of the teachings of Gates.

Accordingly, Gates in view of Atkinson fails to teach or suggest each recited element of claim 24, and the rejection is improper. Applicant respectfully requests that the §103 rejection of claim 24 be withdrawn.

Dependent claims 27 and 30 depend on claim 26, and include all the elements of claim 26. Gates, as discussed above in support of claim 26, fails to teach or suggest the element "generating a policy digest for a cached policy that applies to a client, the policy digest identifying at least one assertion the client is complying with". Atkinson is cited by the Action as to generating or using a hash of the policy; however, Atkinson is of no assistance in light of the teachings of Gates.

Accordingly, Gates in view of Atkinson fails to teach or suggest each recited element of claims 27 and 30, and the rejection is improper. Applicant respectfully requests that the §103 rejection of claims 27 and 30 be withdrawn.

Dependent claim 38 depends on claim 34, and includes all the elements of claim 34. Gates, as discussed above in support of claim 34, fails to teach or suggest the element "extracting at a host a policy digest identifying a cached policy

that applies to a client"; however, Atkinson is of no assistance in light of the teachings of Gates.

Accordingly, Gates in view of Atkinson fails to teach or suggest each recited element of claim 38, and the rejection is improper. Applicant respectfully requests that the §103 rejection of claim 38 be withdrawn.

CONCLUSION

All pending claims 1-38 are in condition for allowance. Applicant respectfully requests reconsideration and prompt issuance of the subject application. If any issues remain that prevent issuance of this application, the Examiner is urged to contact the undersigned attorney before issuing a subsequent Action.

Respectfully Submitted,

Dated: January 14, 2008

By: /Emmanuel A. Rivera/
Emmanuel A. Rivera
Reg. No. 45,760
(512) 344-9931